Comput Syst Sci & Eng (2005) 1: 5–17


International Journal of
Computer Systems
Science & Engineering

# Applications of tool support for risk-informed requirements reasoning

**Martin S Feather, Steven L Cornford, Kenneth A Hicks and Kenneth R Johnson**

Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena CA 91109-8099, USA

Estimating the resources that will be needed to fulfill requirements is an important step. It guides the decision as to whether or not to proceed with development, the selection of which requirements should be sought in that development, and the planning for the development to follow. This paper reports on experience using a tool-supported process for such requirements-time feasibility and resource estimation. This experience has been gathered at NASA and JPL during early-lifecycle planning for space missions. The process has proven successful at identifying problematic requirements (those which will be the most difficult to satisfy), at optimizing the allocation of resources so as to maximize requirements satisfaction, at identifying areas where future research should be focused, and at supporting tradeoff analyses among major alternatives.

Keywords: requirements reasoning, tool support

## 1. INTRODUCTION

At NASA we have been developing and applying a risk management tool, "Defect Detection and Prevention" (DDP). It is based on a simple quantitative model of risk and is supported by custom software. We have used it to aid in planning the development of systems that employ advanced technologies. The tool has proven useful for: quantifying the degree to which a development plan for a technology will satisfy requirements, identifying problematic requirements (those which will be the most difficult to satisfy), optimizing the allocation of resources so as to maximize requirements satisfaction, identifying areas where research investments should be made, and supporting tradeoff analyses among major alternative development plans.

The major elements of the DDP tool have been reported in other publications, e.g. the original concept [1]; look and feel of the custom DDP software [2]; an overview after a couple of years of development and application [3]; an extensive description of DDP's quantitative model of risk [4]. The purpose of this paper is to serve as an up-to-date summary of the *experience* of using DDP.

We begin by relating the requirements-time challenges of space missions to broader requirements engineering concerns across all disciplines, and briefly outline the requirements model that comprises our approach. Then in Section 2 we present the major area of application for our approach, planning for the infusion of promising technologies emerging from research laboratory stage as proof-of-concepts, to ultimately, mission usage. We describe this important but challenging step in the technology development lifetime, and the process we follow for applying our requirements tool to help in this. In Section 3 we step through and illustrate the salient aspects of our approach. Section 4 concludes the paper with a discussion of verification and validation of the overall approach, a discussion of its utility, and its strengths and weaknesses in comparison with related approaches.

### 1.1. Motivation

This paper reports on experience using a tool to support risk-informed requirements reasoning. This experience has been gathered at NASA and JPL during early-lifecycle planning

for space missions. This might at first sound like an esoteric domain, the requirements engineering for which would have little in common with terrestrial applications. While it is true that the domain knowledge itself is atypical (e.g. the temperature environment encountered on the surface of Mars), the concerns that pervade such development efforts are all too familiar ones: juggling scarce resources, novel challenges (either new problems to solve, or old problems to solve in new ways), the need to combine a diversity of knowledge drawn from multiple stakeholders and domains, and finally the need to make influential decisions on the basis of partial and imprecise information. In our setting of space missions, examples of these are as follows: Scarce resources include constrained budgets, schedule pressures forced by celestial mechanics (e.g. launch windows conducive to interplanetary journeys), and severely limited availability of mass, volume and power for physical devices. These have obvious parallels in terrestrial developments, for example time-to-market concerns induce schedule pressures; mass, volume and power constraints arise in the development of handheld electronic devices. The first-time exploratory nature of NASA's space missions induces numerous novel challenges; the desire to maximize the missions' return of science information induces the use of novel technologies, and of existing technologies in novel ways. Parallels to these situations exist in terrestrial developments, for example where new products are envisioned that will themselves create brand new markets. Multiple stakeholders occur in space developments just as they do in terrestrial ones – the scientists whose data will be provided by the missions (in addition, many of these scientists are involved in the development of the instruments that take the measurements), the engineers responsible for the development of the entire spacecraft, the operations team who will control the spacecraft, the long-range visionaries who plan a whole sequence of missions. Development of a spacecraft combines expertise from multiple disciplines (navigation, propulsion, structures, software, etc). Early in the development of a space missions there are many key decisions to be made that will influence the course of the development to follow – the selection of target (e.g. which asteroid to visit, or which landing site to aim for), the determination of spacecraft design (e.g. will it use airbags to cushion its landing?), etc. Terrestrial developments have equally important early lifecycle decisions, such as which partnerships to form, what software architecture to adopt, etc.

## 1.2 Approach

This paper presents our experiences using a risk-informed requirements engineering tool to aid early-lifecycle decision-making. The tool is called "Defect Detection and Prevention" (DDP), the name reflecting its origins as a method intended for quality assurance planning of hardware systems [1].

Successful application of DDP rests upon the combination of: gathering domain experts' information (data from past experience and experts' intuition), following a suitable process (a means to elicit and combine information for the ultimate purpose of guiding experts in their decisions), and utilizing the DDP software tool itself (for representing, reasoning over, and presentation of information). The novel aspects of our approach stem from the underlying process we have developed for this purpose and its custom tool, DDP. The hallmark of the process is quantitative reasoning over a risk-centric model for estimating what it will cost to satisfy requirements. DDP is used to represent the information, perform quantitative calculations, and (through a number of carefully chosen visualizations) convey the information to the human experts throughout the process. A brief summary of this approach follows, sufficient for the purposes of this paper. Readers interested in further details should see [4].

The DDP information model comprises three types of objects: "Requirements" (what it is that the system or technology is to achieve), "Failure Modes" (what could occur to impede the satisfaction of the Requirements), and "PACTs" (what could be done to reduce the likelihood and/or impact of Failure Modes). PACT is an acronym of Preventions, Analyses, process Controls and Tests. These names fit well with our technology assessment studies. In other application areas (e.g. risk management for an ongoing project) we use alternate names for these same concepts, such as "Objectives" in place of "Requirements", "Risks" in place of "Failure Modes", and "Mitigations" in place of "PACTs".

In our quantitative model, Requirements are assigned numerical "weights", indicating their relative importance (e.g. a Requirement with a weight of 10 is twice as important as a Requirement with a weight of 5). Requirements are related to Failure Modes by "Impacts". Each such Impact links a Requirement to a Failure Mode, and has an associated numerical value in the range [0,1], indicating the proportion by which occurrence of the Failure Mode would detract from satisfaction of the Requirement. Each Failure Modes is assigned an "a-priori likelihood", the likelihood that the Failure Mode would occur were nothing done to prevent it. Failure Modes are related to PACTs by "Effects". Each such Effect links a Failure Mode to a PACT, and has an associated numerical value in the range [0,1], indicating the proportion by which the Failure Mode would be reduced were the PACT to be employed (either the Failure Mode's likelihood is decreased, or its Impacts on Requirements are decreased, depending on the kind of PACT). Finally, resource costs are associated with PACTs. Typically a Budget cost value is associated with a PACT. When appropriate, other kinds of costs can be used, such as "Mass", "Power" (electrical power), and "Schedule"; sometimes several kinds of costs are used at once (e.g. a PACT may have both a Budget cost and a Mass cost).

Overall, a DDP model defines a way to calculate the costs and benefits of a proposed development plan. Costs are calculated as the total resource costs of the selected PACTs. Benefits are calculated as the total satisfaction of the weighted Requirements, where the development plan's selected PACTs reduce the extent to which Failure Modes detract from the satisfaction of Requirements.

In each of our studies, the total cost of all identified PACTs far exceeds the resources available. One of the primary uses of DDP is to help identify a cost-effective selection of PACTs. On the occasions when the resources available do not permit a sufficient level of requirements satisfaction, DDP can be used to help reprioritize or downselect from among those requirements, a process called "descoping" [5]. The result can be the decision to discard problematic requirements, relax them (e.g. decrease a quantitative goal

for, say, speed of operation), or deemphasize them (decrease their weight relative to other requirements).

## 2. TECHNOLOGY INFUSION MATURITY ASSESSMENT APPLICATIONS

This section describes the area in which DPP has seen the majority of its applications to date. This area concerns the development and infusion of promising technologies emerging from research laboratory stage as proof-of-concepts, to ultimately, mission usage. Technology research efforts often yield proof-of-concept hardware and/or software demonstrating advantageous features that make them appealing candidates for further development, with the end goal being the use of those technologies on space missions. The next step beyond proof-of-concept demonstration is the development of an "engineering model", i.e. a unit whose design, fabrication and assembly would correspond closely to that of an actual flight unit, and which would then be tested in relevant environments to demonstrate its ability to survive and perform in these environments. Success in these tests would pave the way to use in actual spacecraft missions. The proven design could then be fabricated and qualified at a low additional cost. Since development and testing of an engineering model is a costly undertaking, the decision as to whether or not to go ahead is an important one.

Previous experience has shown that new technologies often stumble at this stage in their development. A smaller proportion than desired succeeds in advancing from prototype to successful engineering model. An informal survey conducted at JPL indicated that the predominant reasons for this are: (1) customer (mission) requirements were miscommunicated, misunderstood, or under-defined, (2) the technology was deemed non-flightworthy in its current state of development (i.e. the technology was subsequently rejected because of some unforeseen engineering issues), and (3) other nearly-equivalent commercially-available technologies could possibly replace NASA-developed technologies.

### 2.1 The Technology Infusion Maturity Assessment (TIMA) process

The JPL "Technology Infusion Maturity Assessment" (TIMA) process has been developed to address these challenges, aiming to clarify the definition of the mission requirements, to identify and address early on the technology-specific engineering difficulties that may result from alternative technology/mission architecture decisions, and to achieve a better understanding of the projected status of the development of competing technologies from the present to the estimated time of delivery.

The TIMA process comprises the following steps:

1. Establish the stakeholders in the technology, i.e. those with the most to gain by infusion.
2. Identify the customer requirements that the technology needs to satisfy before designers and managers will have adequate confidence to infuse the technology into a flight project.
3. Determine the potential, relevant failure modes of the technology and assess how much each failure mode would detract from requirements satisfaction (identifies "tall pole" failure modes).
4. Identify Preventative measures, Analysis, process Controls, and Tests (PACTs) which can reduce the risk of failure, and quantitatively assess the effectiveness of each PACT on each failure mode.
5. Estimate costs of those PACTs as part of an engineering model development and qualification (test) program for the technology in question.
6. Select a set of PACTs that together achieve adequate attainment of requirements (by sufficiently reducing the risks that derive from the identified FMs), taking into account the cost and effectiveness of the individual PACTs. From the results of this PACT selection determine the optimal Cost/Benefit funding recommendations that will improve technology infusion success.
7. Report the TIMA findings to the stakeholders. Include suggested recommendations.

The DDP tool, with its risk-centric model, underpins the TIMA process.

### 2.2. Conducting the technology infusion maturity assessment process

This section describes the way the TIMA process is carried out, and describes DDP's role in support of this process. The TIMA process is conducted by involving all the identified stakeholders at once, together in the same location if possible (on occasion we have made use of teleconferencing to involve off-site members). Co-location allows for ease of communication among the stakeholders, in particular, it allows for the recognition and immediate resolution of points of contention.

The process is subdivided into several sessions, each lasting several hours, and spaced a day or so apart. This is driven by two factors: the difficulty of scheduling longer blocks of time of key experts, and the need to allow time between sessions to seek out specific detailed information that is found to be needed but is not immediately available during the sessions. The nominal process consists of four phases informally referred to as: "*Get to know the technology*", "*Day of the pessimists*", "*Day of the optimists*", and "*Day of the realists*". These are outlined next.

"*Get to know the technology*" – background information is presented on the technology itself, and on the mission needs. Typically this takes the form of slide presentations and handouts, coupled with free form questions and answers, discussions, whiteboard sketches, etc. The purpose of this is to get all parties conversant with the problem area, most importantly, the technology itself and its likely mission application(s). In addition, the TIMA process itself is outlined to the participants, and the DDP tool is briefly demonstrated, but is otherwise not used during this first session.

"*Day of the pessimists*" – the Requirements are elicited and assigned numerical weights indicating their relative importance. The Failure Modes that threaten those Requirements are also elicited, and these two lists are quantitatively linked to each other as DDP's "Impact" links. The word "pessimists" in the title of this session reflects the fact that

the process encourages listing of numerous Failure Modes including novel ones, and standard ones whose solutions are well understood and anticipated. This aspect is in contrast to most other risk assessment methods, which take an existing design and development plan as given and focus their attention on just the risks that remain. We find it useful to gather the information about how standard risks are to be addressed, because the novel aspects (stemming from the new technology and/or from novel mission characteristics) sometimes render the "standard" approaches inappropriate. For example, the novel aspects might change the severity one would normally expect of a Failure Mode, or the effectiveness one would normally ascribe to a PACT. Our process, by forcing the participants to explicitly list such information, encourages the recognition of such instances. It also allows for cost/benefit reasoning – investigation of alternative selections of PACTs. Finally, it serves to document the purpose of PACTs, so that at a future date it will be possible to revisit them, to understand the ramifications skipping some of them, and to understand the purpose(s) of a given PACT as it is applied.

The DDP tool is used in this, and subsequent, sessions, to capture information on-the-fly, to display the accumulated information to the participants, and to guide them in their decision making. We will say more about DDP's use after this overview of the sessions.

Requirements encompass the technology's functional performance requirements (e.g. for a sensor technology such as this, data precision), limits on its resource needs (e.g. bounds on how much power it requires), constraints on its effects on the rest of the spacecraft (e.g. bounds on how much electromagnetic interference it may cause), survivability requirements (e.g. the range of temperatures it must be able to operate within), operability requirements (e.g. its command interface), and relevant development requirements (e.g. constraints on its manufacturability).

"*Day of the optimists*" – the PACTs are elicited (recall that these are potential solutions to the problems represented as Failure Modes) and are quantitatively linked to the Failure Modes they reduce using DDP's "Effect" links. The PACTs are assigned their resource costs. The word "optimists" in the title of this session reflects the fact that the process encourages listing of numerous PACTs, regardless of the sum total of their costs.

"*Day of the realists*" – by this stage the DDP model is complete in the sense that it can be used to compute costs and benefits of alternate selections of PACTs. Using this model the assembled team at this point investigates PACT selections. Either they are able to emerge with an acceptable selection (perhaps with choices among alternative more-or-less equally acceptable selections), or they find that, for the resources available, the Requirements cannot be adequately satisfied. In the latter case, if the problem can be traced to a subset of the Requirements that are proving to be particularly expensive to satisfy, the team can respond by de-emphasizing those problematic Requirements (i.e. making them less important relative to other requirements), weakening them (e.g. substituting a less ambitious quantitative goals that are less impacted by the extant Failure Modes), or even discarding them entirely (which for our technology studies usually corresponds to narrowing the range of applications for which the technology will be suitable). In some cases these insights can be used to discover key problem areas (Failure Modes that adversely impact highly desirable Requirements, for which cost-effective PACTs are lacking) that form the basis for a future line of research.

In practice the partitioning of activities into these phases is not strictly adhered to – for example, if in the middle of costing PACTs, a participant thinks of an important Failure Mode that hasn't already been included, the process would take a detour to added that Failure Mode, link it (by Impacts) to Requirements and (by Effects) to PACTs, before returning to costing the PACTs themselves.

## 2.3 Role of DDP in the TIMA sessions

Use of the DDP tool begins in the "Day of the Pessimists" session and continues thereafter. It is used to capture on-the-fly the TIMA information, to present the status and extent of the information captured so far, to perform calculations over the aggregate of the information, and to help guide the participants in their decision making.

The DDP tool runs as a stand-alone application on a standard windows-based PC, and thus requires only a laptop or desktop PC, and a projector to display the DDP screen image to all (if teleconferencing is used to involve off-site participants, some mechanism to make the DDP screen visible to those off-site participants is necessary). It is important that the DDP screen image be simultaneously visible to all participants. They can view the growing information set, and can suggest new information to be added, offer augmentations and corrections to the existing information, help organize it coherently, etc.

A facilitator, who has some understanding of the technology in question, and also understands the TIMA process and its DDP underpinnings, moderates these sessions. A second individual operates the DDP tool (on occasion we have combined these two roles into one), obviating the need for the stakeholders to master its operation.

Participants see information presented through DDP as soon as it is captured in the tool. This helps them quickly become familiar with DDP's information displays. Ease of understanding is also helped by DDP's employing familiar elements of Microsoft Windows(r) look and feel where appropriate (DDP is implemented in Visual Basic, whose controls offer much of this functionality). Finally, DDP adheres to some simple user interface good practices, for example (mostly) consistent use of color to distinguish different kinds of information. Participants offer feedback on the information they see presented, which the DDP operator incorporates into the evolving DDP model.

An important characteristic of DDP is that data entry of participants' information be rapid. This ensures that there is rarely need to halt the free flow of information to let the tool catch up. Having an operator familiar with the tool is very helpful in this regard, especially when coupled with various tool features that facilitate information entry and reorganization. None of these are particularly groundbreaking (e.g. control key combinations, mouse and menu options that speed data entry and editing), but their inclusion has grown through our own hand-on experiences operating the tool.

The DDP tool is used to both capture information, and

also to calculate results from the aggregate of that information. Presentation of the results of these calculations back to the participants is another important aspect of the tool, for which a number of visualizations have been created. Some of these will be seen in the section to follow.

## 3. EXAMPLES

This section presents examples drawn from applications of DDP in support of various Technology Infusion Maturity Assessment (TIMA) studies.

A typical TIMA was one used to guide the development of a Micro Electrical Mechanical System ("MEMS"). The system was a novel design for a miniature, low power sensor intended for use in space applications (spacecraft themselves, and also surface vehicles, i.e. rovers). The design was the product of a research effort, which had developed a proof-of-concept unit demonstrating its advantageous features, namely high performance (in terms of its sensing capabilities), small size, low mass and low electrical power needs. These characteristics made it attractive in comparison to traditional sensors for the same purpose. The design was at the stage where the next step would be the development of an "engineering model", i.e. a unit whose design, fabrication and assembly would correspond closely to that of an actual flight unit, and which would then be tested in relevant environments to demonstrate its ability to survive and perform in these environments. Development and test of an engineering model is typically a costly undertaking. This motivated following the TIMA process to ascertain whether or not the MEMS technology was suitably mature and appropriate for the intended applications, and to construct a cost-effective development and test plan.

The TIMA involved over a dozen stakeholders whose total experience spanned the domains of: systems engineering, space experiments, avionics, materials, packaging, manufacturing, testing, experimental design, failure analysis, quality assurance, mission technologies, MEMS research, program management, and spacecraft mission (the mission anticipated to be the first user of this MEMS technology, and source of funding for its next stage of development).

During the course of this TIMA, DDP was used to capture the following amounts of information:

- 35 Requirements, of which 29 were determined to be relevant to the study,
- 68 Failure Modes, of which 58 were determined to be relevant to the study,
- 36 PACTs from which to select,
- nearly 700 quantitative Impact links connecting Failure Modes to Requirements, and
- nearly 300 quantitative Effect links connecting PACTs to Failure Modes.

Observation: *This is representative of the quantities of information involved in TIMA studies of individual technologies. This amount of information warrants an organized process for its elicitation, so that the process is efficient (does not waste the valuable time of the participants) and effective (has the breadth and depth required for informed decision making.*

The connectivity between Requirements, Failure Modes and PACTs is shown in Figure 1, a DDP-generated display that draws a row of tiny blue squares along the top, one for each of the Requirements, a row of tiny red squares along the middle, one for each of the Failure Modes, and a row of tiny green squares along the bottom, one for each PACT. Red lines connect Failure Modes to the Requirements they Impact, and green lines connect PACTs to the Failure Modes they Effect. This kind of display vividly reveals the cross-coupled nature of the information typically gathered by TIMA studies. It is evident that a Requirement is typically Impacted by multiple Failure Modes, that a Failure Mode may Impact multiple Requirements, etc. Also remember that underlying each of the links is a quantitative value indicating the proportion of the Impact or Effect. It is this highly cross-coupled aspect that makes early design-time decision making for projects of this nature challenging.

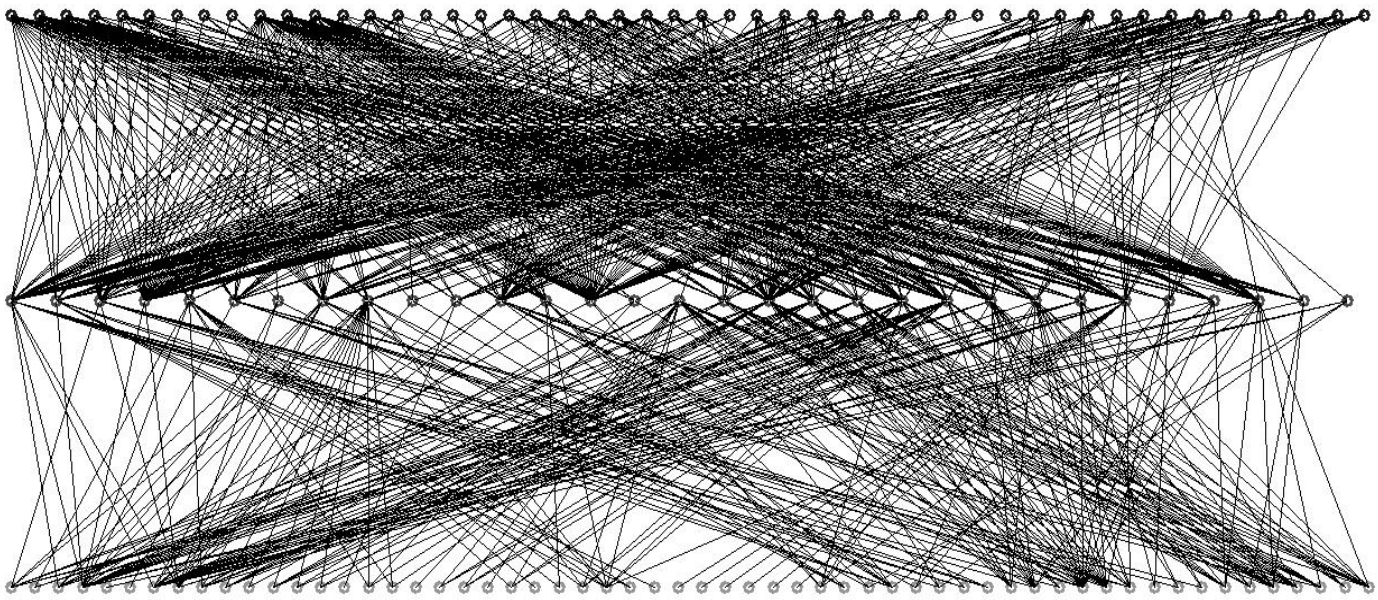

**Figure 1** Connectivity between requirements (top row), failure models (middle row) and PACTS (bottom row)

Observation: *The highly intertwined nature of this study's data is typical of that we find over many TIMA studies. The quantities of data, coupled with its intertwined nature, accounts for the challenges of decision-making in this setting. Tool support is needed to assist in this.*

We have developed our TIMA process and the DDP tool that supports this process so as to address these challenges,

## 3.1 Information elicitation and its presentations

A simple but important step in managing the quantity of information is to organize. Within TIMA sessions, tree structures are used to group information – Requirements, Failure Modes and PACTs – into hierarchies that assist in navigation (rapidly locating where information would be placed) and decompose the elicitation of information into substeps, thus encouraging a more thorough treatment. DDP does not impose any preset hierarchy – rather, whatever structure best suits the study is created on the fly.

Observation: *In our experience we find that determination of a suitable hierarchy is an iterative process. We begin with an initial set of high-level categories, but as the study progresses we can and often do reorganize the hierarchy to emerge with a more suitable structure. Furthermore, we do not insist that tree structures are balanced or expanded to a uniform depth. This freedom permits the DDP representation to capture where the study focuses in greater or lesser detail on various aspects. For example, Requirements that are relatively standard and little impacted by the novel aspects of the technology at hand will not need to be subdivided, whereas those that concern the critical and novel issues will be refined further.*

Familiar tree views are used to visually present these hierarchies. Various menu, mouse and control key operations allow for the rapid rearrangement of hierarchy.

A portion of such a tree view, showing some of the Requirements for one of our studies, is shown in Figure 2. The Requirements are automatically numbered in sequence (the numbering can be changed to tree structured, e.g. 1.3.1, if desired) and their pithy names listed. Checkboxes control which elements are taken into consideration. The column to the left can be set to show an attribute value for each of the items; in this case it is showing the user-assigned "weights" of each of the Requirements.

In the DDP model, Impacts link Failure Modes to the Requirements whose satisfaction they threaten and Effects link PACTs to the Failure Modes they reduce are entered. Quantitative values in the range [0,1] are assigned to these links. Recall that the value of an Impact link indicates the proportion of the Requirement whose satisfaction would be lost were that Failure Mode to occur, while the value of an Effect link indicates the proportion by which either the likelihood or the impact of the Failure Mode would be reduced were the PACT to be employed.

Observation: *For the TIMA studies' purpose of early-lifecycle planning of technology applications and development, it suffices to capture relatively coarse estimates for these quantitative values. For example, since DDP values for these quantities are in the range [0,1], typical values found in TIMA studies are 1.0, 0.9, 0.7, 0.3, 0.1. Such coarse precision commensurate with the precision of information available at these early stages of planning.*

Given $m$ Requirements (or PACTs) and $n$ Failure Modes, there are $m \times n$ possible Impact (or Effect) links between them. This $O(n^2)$ phenomenon means that a significant portion of the TIMA process involves gathering these values from the experts. In practice only a fraction of the possible pairs are links (absence of a link is equivalent to a zero-valued link. For the MEMs study numbers cited earlier (35 Requirements, 68 Failure Modes, 36 PACTs, nearly 700 Impact links and nearly 300 Effect links), there were approximately 30% Impact links out of all possible such links, and approximately 12% Effect links. In another TIMA study, the corresponding numbers were 33% and 16%. When gathering these values it is often easy to know where to focus attention and where to skip over portions that are not going to involve links, so the process is not as daunting as it first seems.

DDP provides several alternate visualizations suited to entering, viewing and editing this linkage information. In addition to the connectivity graph shown earlier in Figure 1, there is a spreadsheet-like matrix view, a tree view where the two trees of items being linked are shown side by side with the values of a selected row and column shown in the cells to their left, a list view which can be thought of as the matrix with its empty cells discarded, and a thumbnail view of the entire matrix. No single one of these has proven ideal in all circumstances; each has its own strengths and weaknesses. For example, the matrix view is good for seeing the "neighborhood" of quantitative values, but must either limit the neighborhood to a small window, or ruthlessly truncate the names of items (this is still the case even though we use "pithy" (i.e. short) names for items); conversely, the two trees are good for showing the names of items, but is limited to display of only one row and one column of data at once. Each of these compromises one of the tenets of good data visualization [6].

Observation: *The fundamental challenge is that there is more information than can conveniently be displayed on a single screen. The preferred visualization depends on the data (visualizations that work well for one study's data do not necessarily work as well for another). Hence the DDP tool offers several visualizations, and the tool operator must know when it is appropriate to switch from one to another. Adopting a methodical process to filling in the data also helps. For example, an approach we often follow when eliciting Impact values is to pick a Requirement, and step through the Failure Modes one by one, eliciting their Impact values on the Requirement.*

Observation: *During the process of eliciting this quantitative data (regardless of interface) dissent is resolved not by voting or averaging, but by delving into the reasons for the dissent. Almost always this is then resolved by refining of the level of detail, e.g. when dissent is over the Impact that a*

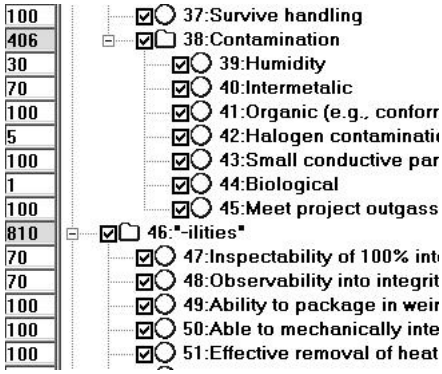


**Figure 2** Tree View (portion of)

*Failure Mode has on Requirements, it may be resolved by decomposing that Failure Mode into several more specific Failure Modes, thus allowing for finer distinctions between their Impacts.*

## 3.2 Calculated information

A hallmark of our approach is its quantitative treatment of information: Requirements are ascribed numerical weights indicating their importance, Failure Modes are ascribed a-priori likelihoods, and PACTs are ascribed costs. The linkages between these are also quantitative: Impacts link Failure Modes to Requirements indicating the proportion of loss of Requirements satisfaction that a Failure Mode would cause, and Effects link PACTs to Failure Modes indicating the proportion by which the PACT would reduce the (likelihood or severity) of the Failure Mode.

Observation: *This quantitative interpretation allows for a number of calculations. As seen earlier, the numerical values are not all that precise, yet we find that the aggregates of these relatively coarse values yield important insights into the overall dataset.*

It is from these insights that we derive guidance for estimating feasibility, descoping requirements (if it is found necessary to do so), and planning the development of how to satisfy those requirements (i.e. in DDP terms selecting from among the PACTs). The most significant of the quantitative calculations that DDP performs are as follows:

*Total Impact on each Requirement*: For each Requirement, its weight times the total Impact on it from all the extant Failure Modes. DDP calculates both the "PACTed" value (i.e. taking into account the beneficial effects of the currently selected PACTs at reducing Failure Modes) and the "unPACTed" value (i.e. without taking PACTs into account). The utility of the "PACTed" calculation is to show the current state of the Requirement's satisfaction, which can be compared to the "UnPACTed" calculation to show the benefit of PACTs.

*Total Impact of each Failure Mode*: For each Failure Mode, the total Impact it has on the weighted Requirements (e.g. if a Failure Mode has an Impact of 0.7 on a Requirement with weight 12, it will accumulate 8.4 units from this Impact). As for the previous item, two versions of this value are calculated: "PACTed" and "UnPACTed". The utility of the "PACTed" calculation is to show the extent to which a Failure Mode is currently detracting from satisfaction of Requirements, which can be compared to the "UnPACTed" calculation to show the benefit of PACTs.

*Total Benefit of each PACT*: For each PACT, this is the total gain of weighted Requirements satisfaction that would be achieved were that PACT to be selected. Two versions of this calculation are performed: the "solo" benefit of the PACT, i.e. the gain obtained if this were the one and only PACT selected, and the "delta" benefit of the PACT, i.e. the gain obtained by this PACT is selected as compared to not selected while taking into account all the other currently selected PACTs. The "solo" value gives an estimate for the sum total effectiveness of a PACT, while the "delta" value shows the benefit of adding it to the current set of selected PACTs. Because of the way that DDP models the combination of multiple PACTs against the same Failure Mode, a phenomenon of "diminishing returns" comes into play which typically renders the "delta" value of a PACT to be less than its "solo" value, so both these calculations have merit.

Observation: *These summary measures are analgous to (indeed, motivated by) the risk importance measures found in PRA. For example the Risk Reduction Worth "... is a measure of the change in risk when a basic event ...is set to zero. It measures amount by which risk would decrease if the event would never occur..." [7]. While PRA measures relate to risks, DDP can also offer measures that relate to PACTs that mitigate risk,, since these are represented explicitly within the DDP model.*

*Total Cost of PACT selections*: For a given selection of PACTs, DDP computes their total cost(s) in the various kinds of resources (e.g. budget, mass, power) utilized in the study at hand.

Observation: *All the calculations are of one of two kinds of values – benefits, derived from the weights the assigned to the requirements, and costs, derived from the costs assigned to PACTs. For example, Impact values are derived from Failure Modes' reductions of Requirements' satisfaction, and thus are measures of (loss of) benefit.*

## 3.3 Visualizations of the results of information calculations

The numerical results of these calculations can be shown as numbers, exported in tables, printed in reports etc. However, for decision-making we turn to cogent visualizations to reveal the non-trivial amount of information in a manner conducive to decision making. Our most commonly used such visualizations are forms of bar charts.

Figure 3 shows DDP's bar chart window displaying the status of some Requirements. Each Requirement is represented by a bar extending horizontally towards the left; the number and name of the Requirement (and number and name of its parent Requirement, etc) is listed to the right of the bar. Three quantitative values are show for each

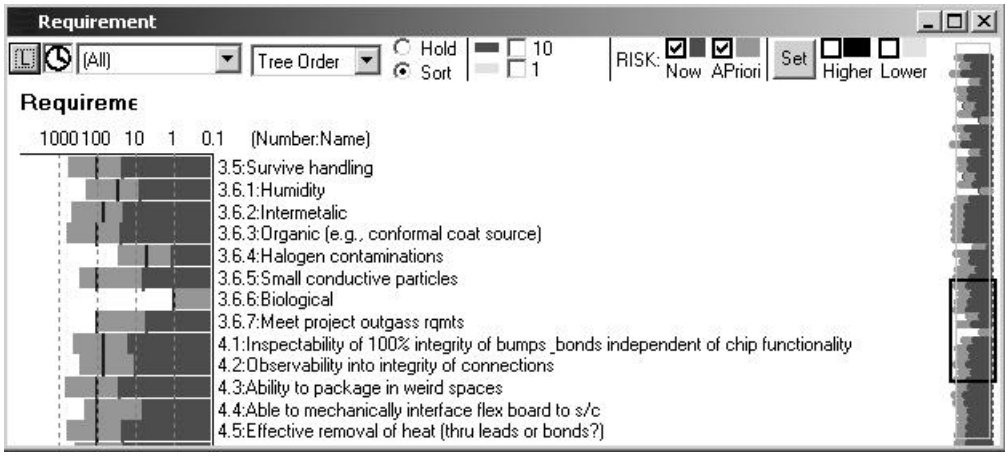


**Figure 3** Bar chart view of the status of some requirements

Requirement: (1) its weight, (2) the extent to which its satisfaction is currently threatened by Failure Modes (taking the current selection of PACTs into account), and (3) the extent to which its satisfaction was originally threatened by Failure Modes (i.e. ignoring PACTs). In the bar chart these three values are indicated as follows:

1. Weight: indicated by a vertical dark line (e.g. the topmost bar's such line is at the 100 level),
2. The extent to which its "PACTed" satisfaction is threatened by Failure Modes (i.e. taking currently selected PACTs into account): indicated by the position of the left end of the darker portion of the bar (e.g, the topmost bar's darker portion extends left to approximately the 20 level - note that a log scale is employed).
3. The extent to which its "UnPACTed" satisfaction was originally threatened by Failure Modes (i.e. ignoring PACTs): indicated by the left end of the lighter portion of the bar (e.g. the topmost bar's lighter portion extends left to approximately the 800 level).

In the interest of brevity in this paper, the window has been sized to show only a portion of the Requirements (the ones listed earlier in Figure 2's tree). In fact, all 50 of this study's Requirements can be presented on one screen. As the number of items increases beyond those that can comfortably presented in a single screen, it becomes necessary to utilize thumbnails and scrolling. This capability is seen to the right of the figure, where a miniature thumbnail shows all 50 Requirements, with those currently in view to the left indicated by the black rectangular outline. Another capability utilized in such circumstances is the ability to sort the items according to some chosen criteria, e.g. for Failure Modes, sort by their "PACTed" total impact.

## 3.4 Insights from visualizations

Bar charts are used to display the status of quantitative attributes of each of DDP's kinds of information, Requirements, Failure Modes and PACTs, as follows:

*Requirements*: the chart of Requirements shows, for each Requirement, its weight, the extent to which its satisfaction is currently threatened by Failure Modes, and the extent to which its satisfaction was originally threatened by Failure Modes. From this chart it is easy to see the extent by which Requirements are threatened by Failure Modes, the relative importance of those requirements, and the requirement-by-requirements benefit that the current selection of PACTs conveys. Observation: Insights from the Requirements chart are useful in requirements descoping decisions, by identifying the problematic requirements to discard, relax (e.g. decrease a quantitative goal for, say, speed of operation), or deemphasize (decrease their weight relative to other requirements). They also show in terms of requirements satisfaction the benefit of the currently selected Mitigations (e.g. what are we getting for the money?)

*Failure Modes*: the chart of Failure Modes shows, for each Failure Mode, its current ("PACTed") total weighted impact on Requirements and its original ("UnPACTed") total weighted impact on Requirements. From this chart it is easy to see which Failure Modes are proving to be the most problematic. They also reveal the Failure Modes proving to be the least problematic (which results when their original impacts on the weighted requirements are small, their a-priori likelihoods are small, and/or when the current selection of PACTs is very effective at reducing them).

Observation: I*nsights from the Failure Modes chart are useful in recognizing when planned effort is mis-allocated (excessive resources are being expended to reduce already trivial Failure Modes, while other much more serious ones are relatively unaddressed). Such mis-allocation can arise when what would normally be a good strategy does not translate well to the novelty of the technology and/or its novel application. The absence of any sufficiently effective set of PACTs to reduce an important Failure Mode can justify a spin-off effort to focus research attention on that problem area.*

*PACTs*: the chart of PACTs shows, for each PACT, the benefit it conveys in terms of reducing Failure Modes (equivalent to increasing Requirements satisfaction).

Observation: *this is the least used of the bar charts. Rather*

*than use this as the guide to selecting PACTs, we tend to work from the Failure Modes, selecting the PACT(s) needed to sufficiently diminish the most important Failure Modes. Instead, we find much more value in a chart that combines the cost and benefit aspects, discussed next.*

Overall, DDP defines a cost-benefit model: costs are calculated as the total resource costs of the selected PACTs; benefits are calculated as the total satisfaction of the weighted Requirements, where the selected PACTs reduce the extent to which Failure Modes detract from the satisfaction of Requirements. Such a model can be used to locate (near) optimal "solutions".

We have built into DDP a heuristic search mechanism (simulated annealing) for this purpose. The search can be used to locate the selection of PACTs that achieve maximum total Requirements satisfaction while costing no more than some cost ceiling, to locate the selection of PACTs that minimize cost yet achieve at least some level of Requirements satisfaction, or a hybrid of the two. By extending this search across the entire cost range it is possible to emerge with a picture of the overall cost-benefit space of the study in question. Figure. 4 shows an example of DDP's visualization of the results of such a search, annotated to indicate the key insights it offers. The black "cloud" consists of some 300,000 solutions, each a selection of PACTs. A given solution has a cost and benefit as calculated by DDP, and based on these values is located with respect to the horizontal axis (cost) and vertical axis (benefit). Thus the points along the upper left boundary of the "cloud" represent optimal solutions at various cost levels (the "Pareto frontier" [8]. We have used the information revealed by these searches to locate the "sweet spot" region where the level of funding is appropriate. In Figure 4 there is one such "sweet spot" region, where the frontier's slope is approximately 45 degrees – at lower level of funding, the benefit drops off dramatically, while at higher levels of funding a law of diminishing returns means that the additional funding achieves only tiny increments in benefit. In some of our studies the frontier has more "kinks" in it, has several such regions. For more details on this, see [9].

Observation: *We find it important to retain the involvement of the human experts throughout the process. Heuristic searches such as those described above are useful for indicating overall patterns and trends, the results of which are made available for further scrutiny by the experts. Our experience suggests that the by blend of automation together with the expertise, insights and guidance of systems engineers is an effective combination.*

## 4. CONCLUSIONS

The obvious questions to ask of our TIMA approach as a whole are ones of:

- verification – how do we check that internal calculations of the DDP software are correct?
- validation – how do we check that the overall results of a TIMA process are correct?
- utility – how do we measure and compare the cost and benefit of performing the TIMA process?
- comparison – how do we compare TIMA to other approaches to early-lifecycle decision making?

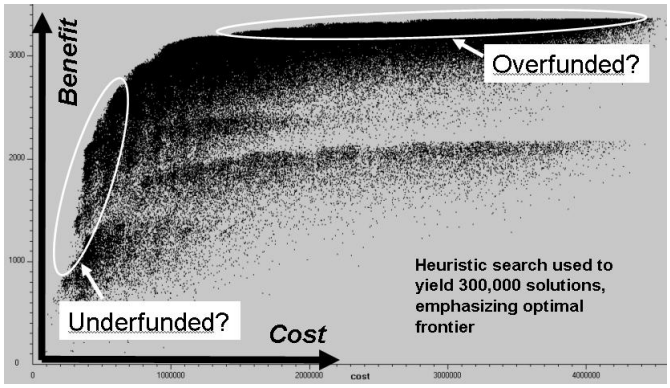


**Figure 4** Cost-benefit space

The subsections that follow address these questions, followed by a brief conclusion subsection.

### 4.1 Verification

We use a mix of techniques to verify correctness of the DDP software's internal calculations:

*Testing* to compare DDP's calculated results against manual "pencil and paper" calculations of the same problems. Generally these are small test cases, because of the tedium of performing the manual calculations.

*Recreating in DDP risk results from other approaches, and comparing those results to the ones calculated by DDP.* For example, we arranged to import information from a spreadsheet implementation of a FMECA; the spreadsheet's calculations of risk priorities could be compared to DDP's equivalent. (We used the worksheet from http://www.fmeainfocentre.com – the "FMEA Info Center" – see [10] for details of this experiment.)

*Comparing risk calculations done by other programs.* We are in the process of connecting DDP with Galileo [11], a Probabilistic Risk Assessment tool. This allows us to transfer simple fault trees (a recent extension to DDP's Failure Modes) from DDP to Galileo, and have the latter perform the calculations of likelihoods in place of DDP. We can compare the results to those that DDP's own calculations would yield. (The longer term purpose is to utilize Galileo's capabilities to perform computations over more elaborate fault trees, computations for which no DDP implementation exists.)

*Comparing results computed current DDP* against those computed by earlier versions of DDP on datasets created when those earlier versions were used in TIMA studies. Those earlier studies give us non-trivial test cases. This is feasible because we make DDP's successive releases "upwards compatible" (i.e. DDP can read a dataset created by any earlier version of DDP).

*Comparing results computed current DDP against those computed by an alternate implementation of DDP.* For a while there was an alternate implementation in Java rather than Visual Basic of portions of DDP, including its calcula-

tions. We were able to compare results computed by both implementations on the same TIMA datasets.

*Running internal consistency checks* that values are in expected ranges (e.g. probability values remain at or above zero).

*Comparing internal results computed by old and new code.* When changes are made to (supposedly) speed up the implementation of a calculation we sometimes keep available the old code so as to run in "debugging" mode for a while, during which time we compute internal values using both the old and the new implementation, comparing the two.

*Comparing results that alternate heuristic methods yield.* Several different heuristic optimization methods have been applied to searching for near-optimal solutions within the same large datasets (see section 3.4). The DDP distribution includes an implementation of simulated annealing, but we have also experimented with genetic algorithms, and with machine learning [12]. Such heuristic methods do not necessarily find the exact same results, but they do indicate that they each find the same neighborhood of near-optimal solutions, albeit with different rates of convergence.

Taken together, the above give us high confidence in the correctness of DDP's calculations.

## 4.2 Validation

Validation, checking that the overall results of a TIMA process are correct, is much harder to perform than verification of DDP's internal calculations.

Ideally we would like to validate that the decisions that the TIMA process leads people to make are correct. For example, if the TIMA process leads to the decision to utilize a certain subset of PACTs in order to progress with the development of a technology, we would like to know that those PACTs are the right things to do - that they are both necessary and sufficient: "necessary" in the sense that there is no alternative selection of PACTs that could do better for the same cost, or do the same for less cost, and "sufficient" in the sense that they lead to acceptable satisfaction of Requirements. Actually things are not quite this straightforward – because of the need to predict likelihoods of future events, there is a necessarily a probabilistic aspect. Validity could not be ascertained from monitoring the progress of a single study's technology, but would require monitoring multiple such studies, and gathering statistical evidence. A further complication is that the TIMA process need not be perfect – it can still have utility provided only that it does sufficiently better than alternative decision processes! In our efforts to date we have not gathered enough data to draw any conclusions along these lines.

A possible experimental approach to validation would be to do a TIMA study and independently run some alternate decision making process on the exact same problem. To achieve independence we would have to find two disjoint sets of experts to be involved in each study. To date we have not had the luxury of the time, money and personnel necessary to perform such a study.

In place of the above, we rely on the following observation that gives us partial confidence in the validity of the TIMA process:

In applications of the TIMA process to study technology infusions, we find that the intermediate findings of the TIMA process (e.g. determination of which Failure Modes are the cause of the greatest loss of Requirements satisfaction) usually agree with the intuitions of the technology experts. This leads to decisions that, on the whole, mirror those that the experts would have made anyway. (We base these assertions on the comments the experts make during and after TIMA studies, and on the reactions we get to TIMA studies' recommendations from other experts not involved in the studies themselves.) We stress that not all the elements of a TIMA study match the experts' intuition – in many of the studies, there is something surprising about the findings. In these cases we find that the experts are at first skeptical, but upon closer inspection become persuaded that the findings are correct. Closer inspection is achieved by investigating why the quantitative model of Requirements, Failure Modes and PACTs points to those findings; such investigations are something that the DDP software readily supports. For example, if the TIMA session suggests that a Failure Mode is more severe than expected, the underlying DDP calculations can be queried – which significant Requirements does that Failure Mode threaten, and why isn't it much mitigated by the selected PACTs? The conclusion of this is usually agreement by the experts that the TIMA process has uncovered something correct, but which they might not otherwise have recognized so early in the technology infusion lifecycle. In fact this recognition of surprises is the primary benefit of the TIMA process, which we discuss in the next section.

## 4.3 Utility

The utility of the TIMA process depends on both its *cost* and *benefit*:

We measure a TIMA study's *cost* in terms of the total person-time it takes to run the TIMA sessions. The number of experts involved in a technology infusion TIMA study has ranged from as few as half a dozen to as many as fifteen, plus the participation of a facilitator and, in some cases, a separate person to "drive" the DDP tool. It is typical to need four half-day sessions to gather the data and make decisions, so the total cost can range from $8 \times 4 \times 4 = 128$ hours to $17 \times 4 \times 4 = 272$ hours. In addition there is usually the need to spend another 10–20 hours to compose a final report documenting the study. These are not trivial amounts of time, especially given that we seek the involvement of experts in the various discipline areas, experts whose valuable time is in continual demand. However, the cost of development and testing of an engineering model can be several millions of dollars, and span several years, so in relative terms a TIMA study is not a huge expense.

The *benefit* of a TIMA study is less easy to quantify than the cost. As mentioned in the previous subsection, the primary benefit is early recognition (earlier than would be the case without having performed the TIMA study) of something of significance with respect to technology infusion decision making. We list several of the more dramatic such findings:

A TIMA study of an advanced storage technology revealed *problematic (at risk) overly stringent requirement*, whose removal permitted dramatic cost and time savings. The technology development was threatened with cancellation, but following the study became proposal-winning concept. The main benefit derived from having requirements

honed to requisite level of mission specificity. The removal of that overly stringent requirement enabled design savings estimated to exceed $1 million. Given that prior to the TIMA study the technology development was threatened with cancellation, a correct quantification of the benefit would be the *lesser* of whatever had been invested in the technology development to date, and the $1M+ design savings, even after allowing for the cost of the TIMA study itself.

A TIMA study's *risk-informed redesign* of a flight experiment systems architecture led to power needs decreased by 68%, mass decreased by 13%, cost decreased by 9%, and the major category of Failure Modes changed from architectural to well-understood design. This was achieved when the TIMA study showed that the existing design was overly focused on resolving relatively minor issues, while more major issues remained outstanding. Once this was recognized, the technology experts were quickly able to see an alternate design that would be an improvement. Again, the design savings were in excess of $1M.

The TIMA study of the Micro Electrical Mechanical design for a miniature, low power sensor intended for use in space applications, discussed briefly in Section 3, led to the following positive outcomes:

- Emerged with clear definition of work needed to mature the technology
- Identified a commercialization opportunity based on its unparalleled performance
- Improved initial estimates of cost-to-completion by considering key tasks that were flight project specific in nature
- Arrived at design that minimizes risk specifically for the flight project (the study revealed a critical design choice that if made a particular way led to a significant payoff by eliminating several complicated fabrication and data processing issues.

A TIMA study of adaptation of GUI-driven autocoding to run as flight instrument controller was used to *achieve sufficient understanding of the risks to enable a decision to "go ahead*" with the project. Prior to the TIMA study the benefits were well understood, but there was concern that serious risks might be unfamiliar to the technologists; the TIMA approach *helped identify risks* (in DDP terminology, "Failure Modes", e.g. unrelocatable code) and PACTs to adequately mitigate those risks. The result was a decision to go ahead with the use of this technology, and also revealed there to be a business application area for this technology outside of its usual domain.

Although we cannot accurately quantify the benefit of each of the above cases, we are confident that it well exceeds the cost of the TIMA studies in question. Initial TIMA studies were supported by research funds. Those early studies confirmed the viability and value of the TIMA process. Since then, studies are paid for by the spacecraft technology development funds. This also lends credence to the claim that the TIMA process is of net benefit.

## 4.4. Comparison with related work

There are many other approaches that support decision making early in the development lifecycle. We point out similarities between some of those other approaches and the DDP-supported TIMA process. We then focus on some specific differences between TIMA and aspects of other approaches.

### 4.4.1 Similarities

The "cost-value" approach to prioritizing requirements in [13] showed the value of estimating feasibility (cost) as well as benefit when deciding upon requirements. At the heart of their approach is a cost-value diagram, which plots each requirement's relative value and implementation cost, facilitating the selection of an appropriate subset of requirements.

The WinWin project [14] supports multiple stakeholders to identify conflicts between their respective evaluations of requirements, and to locate feasible solutions that are mutually satisfactory combinations of requirements. The WinWin approach is supported by a custom tool, the benefits of which are reported in [15] – our combination of experts, process and tool very much echoes their experience.

The relative simplicity of the DDP model gives it the ability to span a wide range of concerns (e.g. technical and programmatic risks). In this respect it is reminiscent of, for example, the openness of the i* model [16, 17], used for evaluation of (among other things) software design alternatives.

Overall DDP is reminiscent of QFD (Quality Function Deployment) method [18], widely used across a range of industries and application areas. DDP has a more quantitative, risk-centric perspective, with a probabilistic interpretation pervading its Requirements-FailureModes-PACTs model.

### 4.4.2 Limitations and contrasts

The DDP model schema (notably the formulae by which impacts of multiple Failure Modes on the same Requirement "add up", and by which multiple PACTs against the same Failure Mode combine) is inflexible. These formulae are preset, and do not necessarily apply well to all situations. While there are some workarounds that can be employed if need be (e.g. representing a combination of PACTs as a distinct PACT whose effectiveness at reducing risks can be asserted), they are clumsy to use. Other researchers adopt models that can be constructed to match the case at hand, and thus more faithfully represent the software development process, e.g. the Bayesian Belief Net models of [19], or the simulation models of [20].

DDP lacks a means for validation of its models. The aforementioned formulae were chosen to be plausible, but are not based on a solid body of evidence. Likewise the experts' estimates that comprise a significant portion of most DDP models are constructed on-the-fly, and so do not have an explicit pedigree to experiential data. This is in contrast to software estimation techniques such as COCOMO and, more recently, COQUALMO and iDave [21, 22] which are derived from data from past software projects, possibly tempered by a consensus process of experts (e.g. using Delphi techniques). Also similar is the stochastic model of [23].

There is still an "art" to populating a DDP model. The use of pre-populated models as a starting point (e.g. a risk taxonomy of generic software development problems) is somewhat helpful, both to establish an overall structure, and to serve as a reminder (much like a checklist). However, there is need

for discretion over how much additional information to add (what is the scope of the study?), and over how much detail to descend to (e.g. does the single Mitigation "software inspections" suffice, or is there need to distinguish among alternative forms of inspections, e.g. Fagan inspections, Perspective Based Reading, etc.). The danger of staying at too narrow a scope and/or too high a level is lack of coverage and discrimination among significantly distinct cases, while the danger of overly broadening the scope and/or descending to too low a level is the increased effort it takes to populate the model. We address this problem by using a DDP-knowledgeable person to facilitate the meeting (in fact, we usually use two such people – one to serve as facilitator, the other to "drive" the DDP software).

DDP's probabilistic model is overly simplistic when compared to the structures seen in use in full-fledged Probabilistic Risk Assessment [7]. For example, PRA tools such as Saphire, QRAS and Galileo have explicit notions of temporal dependencies (through event sequence diagrams or phase-dependent fault tree gates), and of probability distributions with which to capture and reason about uncertainties. Explicit treatment of uncertainty is also seen in, for example, the Accord system of Robust Designs [24]. Uncertainty reasoning is something we are beginning to add to DDP. We recently incorporated logical fault trees into DDP's Failure Mode structures [10], as a small step toward the full representational power of PRA. We have explored the use of DDP as an agile precursor step that serves to indicate where a more elaborate PRA model needs to be constructed [25]. In the software arena, however, it is less routine to apply PRA methods; instead, modest applications of fault trees, Software FMECAs and hazard analyses (and combinations of them, e.g. [26]) are the norm. For such applications DDP seems to be relatively well suited. Overall, DDP's model has proven adequate to support the TIMA process in application to studies of technology infusion, as described in this paper.

In the software arena, DDP is reminiscent of the KAOS models of [27], specifically their treatment of "obstacles" (akin to DDP's "Failure Modes"). However, whereas the KAOS models emphasize a logic-structure based treatment (including temporal logic to capture, e.g. timing requirements), DDP has emphasized instead a quantitative treatment more immediately suited to capture and tradeoff reasoning over and between so-called "non-functional requirements". It is intriguing that in recent work the KAOS approach is being extended to incorporate quantitative reasoning [28], while, as mentioned above, logical fault tree constructs are being incorporated into DDP. It seems these alternate approaches are each expanding in the direction of the other.

## 4.5 Conclusions

This paper presented our experience of applying the DDP-supported TIMA process in aid of technology infusion studies. We summarized the primary features of the TIMA process and DDP, its software support. We also discussed the issues of verification, validation and utility of the process, and made comparisons with other approaches that support decision making early in the development lifecycle.

Overall we believe the applications to date indicate the successful nature of the TIMA process. We attribute its success to its blend of three key elements:

- People – getting the experts whose combined experience spans all the discipline areas that are involved.
- Process – an organized approach to eliciting, scrutinizing and utilizing the information elicited from the experts, and keeping them continually engaged in the decision-making.
- Automation – use of appropriate software to represent the non-trivial quantities of information involved, pool that information (in our case, via various quantitative calculations), and present the results using cogent visualizations.

Our work takes place in the context of spacecraft development, however the key factors that drive the need for an approach such as this extend beyond spacecraft development. These factors are:

- Cross-disciplinary concerns that are cross-coupled and interact in multiple ways, as a result of which no one person has expertise that spans all the disciplines, or can simultaneously juggle all the factors involved in large and complex designs.
- Severe constraints on the systems being developed and on the development process itself (schedule and budget pressures, limited operational resources for the deployed system).
- Critical systems for which the cost of failure is high. Such systems must operate correctly in only partially understood environments, and where repair is costly or impossible.
- Unknowns: past experience provides only a partial guide when new applications are to be enhanced and enabled by new technologies of which past experience is lacking.

These are factors that recur in the development of many novel technologies, so we feel that out approach has merit in the early stages of development of many technologies.

## ACKNOWLEDGEMENTS

The research described in this paper was done at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Development of the TIMA process and the DDP software has been supported by NASA's Code Q funded Failure Detection and Prevention Program (FDPP), the Code Q funded Advanced Risk Reduction Task (ARRT) managed by the NASA IV&V facility, and the Code R (recently Code T) funded Engineering for Complex Systems Program. Over several years the insights and guidance of JPLers Chester Borden, John Kelly Timothy Larson, Kelly Moran, Steve Prusha, Andrew Shapiro and Burton Sigal have been most useful in helping us formulate our ideas and bring them to fruition.

## REFERENCES


1 **Cornford, S L** Managing Risk as a Resource using the Defect Detection and Prevention process. *4th International Conference on Probabilistic Safety Assessment and Management*, New York City, NY, International Association for Probabilistic Safety Assessment and Management (September 1998)

2 **Feather, M S, Cornford, S L and Gibbel, M** Scalable Mechanisms for Goals Interaction Management, *4th IEEE International Conference on Requirements Engineering*, Schaumburg, Illinois, IEEE Computer Society, pp 119–129 (June 2000)

3 **Cornford, S L, Feather, M S and Hicks, K A** DDP – A tool for life-cycle risk management, *IEEE Aerospace Conference*, Big Sky, Montana, pp 441–451 (March 2001)

4 **Feather, M S and Cornford, S L** Quantitative risk-based requirements reasoning, *Requirements Engineering* (Springer), Vol 8 #4, pp 248–265 (2003); published online 25 February 2003, DOI 10.1007/s00766-002-0160-y.

5 **Feather, M S, Cornford, S L and Hicks, K A** Descoping. *27th NASA IEEE Software Engineering Workshop*, Greenbelt Maryland (December 2002)

6 **Tufte, E** *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut (1983)

7 **Stamatelatos, M *et al.*** Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners, http://www.hq.nasa.gov/office/codeq/doctree/praguide.pdf Office of Safety and Mission Assurance, NASA Headquarters, Washington, DC 20546 (August 2002)

8 **Sen, P and Yang, J-B** *Multiple Criteria Decision Support in Engineering Design*, Springer-Verlag (1998)

9 **Feather, M S, Kiper, J D and Kalafat, S** Combining Heuristic Search, Visualization and Data Mining for Exploration of System Design Spaces, *Proceedings of INCOSE 2004* (14th International Symposium), Toulouse, France, (June 2004)

10 **Feather, M S** Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface, *Proceedings of the 15th IEEE International Symposium on Software Reliability Engineering*, Saint Malo, Bretagne, France, 2–5 (November 2004). Available from: http://eis.jpl.nasa.gov/ ~mfeather/Publications.html

11 **Sullivan, K J, Dugan, J B and Coppit, D** The Galileo Fault Tree Analysis Tool, *Proceedings of the 29th International Conference on Fault-Tolerant Computing (FTCS-29)* (1999)

12 **Cornford, S L, Feather M S, Dunphy J R, Salcedo, J and Menzies, T** Optimizing Spacecraft Design – Optimization Engine Development: Progress and Plans, *IEEE Aerospace Conference*, Big Sky, Montana, pp 8-3681–8-3690 (March 2003)

13 **Karlsson, J and Ryan, K** A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, pp 67–74 (September/October 1997)

14 **Boehm, B, Bose, P, Horowitz, E and Lee, M** Software Requirements as Negotiated Win Conditions, *Proceedings 1st International Conference on Requirements Engineering*, Colorado Springs, Colorado, pp 74-83 (1994)

15. In, H, Boehm, B, Rodgers, T and Deutsch, M "Applying Win-Win to Quality Requirements: A Case Study", *Proceedings 23rd International Conference on Software Engineering*, Toronto, Ont., Canada, pp 555-564 (2001)

16 **Chung, L, Nixon, B A, Yu, E and Mylopoulos, J** *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, Boston (1999)

17 **Mylopoulos, J, Chung, L, Liao, S, Wang, H and Yu, E** Exploring Alternatives during Requirements Analysis, *IEEE Software* 18(1) pp 92–96 (2001)

18 **Akao, Y** *Quality Function Deployment*, Productivity Press, Cambridge, Massachusetts (1990)

19 **Fenton, N, Marsh, W, Neil, M, Cates, P, Forey, S and Tailor, M** Making resource decisions for software projects, *Proceedings of the 26th International Conference on Software Engineering*, 2004, May 23–28, pp 397– 406 (2004)

20 **Wakeland, W, Martin, R H and Raffo, D** Using Design of Experiments, Sensitivity Analysis, and Hybrid Simulation to Evaluate Changes to a Software Development Process: A Case Study, *Proceedings of International Workshop on Software Process Simulation and Modeling* (ProSim'03), Portland, OR (May 2003)

21 **Boehm, B *et al*.** *Software Cost Estimation with COCOMO II*, Prentice Hall, Upper Saddle River, NJ (2000)

22 **Boehm, B, Huang, L, Jain, A and Madachy, R** The ROI of Software Dependability: The iDave Model, *IEEE Software,* 12(3): pp 54–61, (May/June 2004)

23 **Stutzke, M A and Smidts, C S** A Stochastic Model of Fault Introduction & Removal During Software Development, *IEEE Transactions on Reliability*, 50(2): pp 184–193 (June 2001)

24 **Ullman, D G** *12 Steps to Robust Decisions: Building Consensus in Product Development and Business*, Trafford Publishing, ISBN 1-55212-576-9 (2001)

25 **Cornford, S L, Paulos, T, Meshkat, L and Feather, M S** Towards More Accurate Life Cycle Risk Management Through Integration of DDP and PRA, *IEEE Aerospace Conference*, Big Sky, MT, pp 2.1106–2.1200 (March 2003)

26 **Lutz, R and Woodhouse, R** Requirements Analysis using Forward and Backward Search, *Annals of Software Engineering, Special Volume on Requirements Engineering*, 3, pp 459–475 (1997)

27 **van Lamsweerde, A and Letier, E** Integrating Obstacles in Goal-Driven Requirements Engineering, *ICSE98 – 20th International Conference on Software Engineering*, IEEE-ACM, Kyoto (April 1998)

28 **Letier, E and van Lamsweerde, A** Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering, to appear in the *Proceedings of ACM/SIGSOFT 2004/FSE-12*, Newport Beach, CA (2004)